2.41 DETAILED RESULTS FOR MISSILE MOVEMENT

The Missile Movement Functional Element (FE) for ESAMS 2.6.2 contains a major change as a result of a recent Model Deficiency Report (ESAMS 2.6.2 Street MDR #1, see Appendix B of ASP-I) which directed that a critical line of code be commented out. The line in question sets the flag which controls switching to the alternate Euler angles used to describe the missile's orientation when its pitch angle in near vertical. This action, described in the MDR as a short-term solution, was taken because the use of alternate Euler angles was resulting in unacceptable missile trajectory information. The impact of the short-term solution makes inaccessible all the code which deals with switching to and using alternate Euler angles. The code which is effected remains in the model because the long-term solution presented in MDR #1 requires the alternate Euler angle method be used should the implementation difficulties be corrected. Hence, this section addresses all the code within the FE whether currently used or not. Implementing the short-term solution (excluding the use of alternate Euler angles); however, presents problems as well. Without alternate Euler angles, potential overflow problems exist and incorrect missile trajectory and orientation information is still produced when the missile has transitioned a near vertical orientation. Thus, this FE is incorrect with or without using the code for alternate Euler angles.

One other minor code discrepancy was uncovered which allows a variable to be outside its specified limits for one iteration. The overall code quality is adequate, but a few improvements were recommended. The internal documentation also could be improved, especially in Subroutine ACCEL. External documentation is inadequate.

The table listed below summarizes the desk-checking and software testing verification activities for each subroutine in the Missile Movement FE. The two results columns contain checks if no discrepancies were found. Where discrepancies were found, the desk check results column contains references to discrepancies listed in Table 2.41-4, while the test case results column lists the number of the relevant test case in Table 2.41-6. More detailed information on the results is recorded in these tables.

TABLE 2.41-1. Verification Matrix for Missile Movement.

DESIGN ELEMENT	CODE LOCATION	DESK CHECK RESULT	TEST CASE ID	TEST CASE RESULT
41-1 Compute Inertial Accelerations	ACCEL 116-118	√	41-1	√
41-2 Compute Body Angular Accelerations	ACCEL 126-128	√	41-2	√
41-3 Calculation of 1st and 2nd Derivatives of Primary Euler Angles	EULDRV 81-87	D1	41-7,8, 9, 10	41-7,8,9
41-4 Calculation of 1st and 2nd Derivatives of Secondary Euler Angles	EULDRV 95-101	D2	41-11, 12	√

TABLE 2.41-1. Verification Matrix for Missile Movement. (Contd.)

DESIGN ELEMENT	CODE LOCATION	DESK CHECK RESULT	TEST CASE ID	TEST CASE RESULT
41-5 Convert Euler Angles to Alternate Set	ACCEL 134-141 147-152 BXIEUL 76-78 93-95 UPDATE 129-137	D3 D4	41-3,4 5,6,14	41-3
41-6 Integrate to Find Inertial Positions and Euler Angles	EULDRV 88-93 UPDATE 140-143 148-150	D5	41-15, 17, 20	41-20
41-7 Integrate to Find Inertial Velocities and Angular Rates	UPDATE 144-146 152-154	√	41-16, 18	√
Initialize Double Precision Array	UPDATE 99-123	√	41-13	√
Update Variables and Counters	UPDATE 126,156-172	√	41-15, 16,17, 18,19	√
Maintain Continuity of Missile Trajectory (using complete model)	---	---	41-21	41-21

2.41.1 Overview

The importance of missile movement is its direct relationship to whether or not the missile will intercept the target. The movement of any physical object is caused by the application of forces which cause it to accelerate proportionally to its mass. To simulate how the application of forces accelerates and moves the missile requires an accurate representation of the forces and moments acting on it, the missile mass and moments of inertia. Applying these forces and moments in discrete time steps enables the construction of the missile's trajectory and orientation in inertial space.

The missile's position is represented by its x, y, and z coordinates in inertial space. Its orientation in inertial space is defined by its Euler angles, θ, ψ, and φ. Updating the position requires the 1st and 2nd derivatives of its position (i.e., it's velocity and acceleration) be known. Updating the orientation requires determining the 1st and 2nd derivatives of the Euler angles which include terms for body angular rates and accelerations. When the missile is near vertical, an alternate definition of the Euler angles is required due to potential division by 0 problems.

ESAMS implements the determination of missile linear and angular accelerations and controls the switching of Euler angles in Subroutine ACCEL. Subroutine BXIEUL computes the alternate set of Euler angles after the switch has been made. Subroutine EULDRV computes derivatives of Euler angles. Subroutine UPDATE computes position, Euler angles, velocity, and body angular rates at each time step.

TABLE 2.41-2. Subroutine Descriptions.

MODULE NAME	DESCRIPTION
ACCEL	Computes body and inertial linear accelerations and body angular accelerations. Determines if switch from primary to secondary or secondary to primary Euler angles is required.
BXIEUL	Computes either primary or secondary Euler angles from the current attitude matrix if Euler angles switch in either direction.
EULDRV	Calculates first and second derivatives of either primary or secondary Euler angles.
UPDATE	Initializes current position, velocity, Euler angles, and angular rates. Integrates position, Euler angles, velocity and angular rates, increments integration step counter and autopilot time.

2.41.2 Verification Design Elements

Design elements defined for the Missile Movement FE are listed in Table 2.41-1; they are fully described in Section 2.41.2 of ASP II. A design element is an algorithm that represents a specific component of the FE design. Design element 41-1 computes inertial accelerations from body directional forces. Design element 41-2 computes body angular accelerations using applied moments and missile moment of inertia (which changes with time). Design elements 41-3 and -4 compute the first and second derivatives of the primary and secondary Euler angles for use in integrating Euler angles. Design element 41-5 converts Euler angles to the alternate set after a switch has occurred. Design elements 41-6 and -7 compute the missile's position, orientation, velocity, and angular rates each iteration.

TABLE 2.41-3. Missile Movement Design Elements.

SUBROUTINE	DESIGN ELEMENT	DESCRIPTION
ACCEL	41-1 Compute Inertial Accelerations	Calculate a_{Ix}, a_{Iy} and a_{IIz}
ACCEL	41-2 Compute Body Angular Accelerations	Calculate $\dot{r}$; $\dot{p}$, $and \dot{q}$.
EULDRV	41-3 Calculation of 1st and 2nd Derivatives of Primary Euler Angles	Calculate $\dot{\theta}, \dot{\psi}$ and $\dot{\varphi}$ and $\ddot{\theta}, \ddot{\psi}$ and $\ddot{\varphi}$
EULDRV	41-4 Calculation of 1st and 2nd Derivatives of Secondary Euler Angles	Calculate $\dot{\theta}_a, \dot{\psi}_a$ and $\dot{\varphi}_a$ and $\ddot{\theta}_a, \ddot{\psi}_a$ and $\ddot{\varphi}_a$.
BXIEUL	41-5 Convert Euler Angles to Alternate Set	Calculates alternate Euler angles based on current attitude matrix.
UPDATE	41-6 Integrate to Find Inertial Positions and Euler Angles	Compute current X, Y and Z and θ, ψ, and φ from velocities and accelerations.
UPDATE	41-7 Integrate to Find Inertial Velocities and Angular Rates	Compute current V_{IIx} V_{Iy} and V_{Iz} and r, p and q from linear accelerations and body angular accelerations.
UPDATE	Initialize Double Precision Array	Assign initial values for position, Euler angles, velocities and body rates to double precision array.

TABLE 2.41-3. Missile Movement Design Elements. (Contd.)

SUBROUTINE	DESIGN ELEMENT	DESCRIPTION
UPDATE	Update Variables and Counters	Reassigns variables for position, Euler angles, velocities and body rates. Increments autopilot and step counters.
---	Maintain Continuity of Missile Trajectory (using complete model)	Ensures Missile Movement FE maintains continuity of trajectory information within framework of complete model

2.41.3 Desk Checking Activities and Results

The code implementing this FE was manually examined using the procedures described in Section 1.1 of this report. Any discrepancies discovered are described in the table below.

TABLE 2.41-4. Code Discrepancies.

DESIGN ELEMENT	DESK CHECK RESULT
41-3 Calculation of 1st and 2nd Derivatives of Primary Euler Angles	D1. With the code as written, the secondary Euler angles are never used (see D2). This creates a potential overflow problem in the calculation of 1st and 2nd derivatives of some primary Euler angles, specifically PSID, PHID, PSIDD and PHIDD.
41-4 Calculation of 1st and 2nd Derivatives of Secondary Euler Angles	D2. As currently coded, this design element is not used.
41-5 Convert Euler Angles to Alternate Set	D3. As currently coded this design element (all of subroutine BXIEUL) is not used. D4. The line of code which sets the Euler angle switching flag, IULR, to 1 is commented out. This causes several problems: 1. Potential overflow in the calculation of 1st and 2nd derivatives of primary Euler angles (see D1) 2. Missile can pitch to greater than 90° with errors in heading 3. All code associated with secondary Euler angles, including the subroutine BXIEUL is never used (see D2 & D3).
41-6 Integrate Inertial Positions and Euler Angles	D5. The code which is intended to limit the value of PSI between 0 and 2p does so by altering the value of PSID and not simply resetting PSI within 0 to 2p when PSI is updated. Also, the location where this occurs in the code allows PSI to be outside the 0 to 2p range for one time step.

Except as noted in Table 2.41-5, overall code quality and internal documentation were evaluated as good. Subroutine I/O were found to match the ASP II descriptions. The major code change in Subroutine ACCEL results in significant differences between the logical flow presented in the ASP II documentation and that which is currently followed by the code. All pathways and algorithms relating to the switching of Euler angles are addressed in the ASP II logical flow and are present in the code. But, due to the critical line of code being commented out, these pathways and algorithms are not accessed in the code. Figure 2.41-1 is the logical flow chart presented in the ASP II document with the shaded blocks showing the pathways and algorithms made inaccessible by the MDR-directed code change.

TABLE 2.41-5. Code Quality and Internal Documentation Results.

SUBROUTINE	CODE QUALITY	INTERNAL DOCUMENTATION
ACCEL	As per MDR #1, the critical line of code which controls switching of Euler angles is commented out (line 137) causing a substantial change in the operation of Missile Movement FE and the trajectory and orientation data calculated by the model as a whole.	1. The Purpose states that “BXIEUL is called to compute the body to inertial transformation matrix.” As coded, BXIEUL is never called. BXIEUL computes the alternate set of Euler angles using the current attitude matrix. The switching of Euler angles is not discussed in the Purpose. 2. Variable D2R75 is given a data value but is listed only in a comment line and is also listed as a local variable. 3. Units for thrust are Newtons, not kg. 4.The comments on lines 135 and 139 do not match the actual code for addressing the conditions under which Euler angles switch. 5. The comment “determine need for switch” on line 143 should read “perform switch”.
BXIEUL	OK.	The reason for the IF statement on line 75 is not explained. The ELSE condition of this statement is used only when BXIEUL is called from VSTATE to force computation of primary angles for printing purposes only.
EULDRV	There is a potential for overflow problems when secondary Euler angles are not used. The method and location of limiting PSI to within 0 to 2π is confusing and improper. Otherwise OK.	The code which limits PSID should be explained in a comment.
UPDATE	OK	The header says “values for the Euler angles are set to the primary angles”, primary should be “current”.

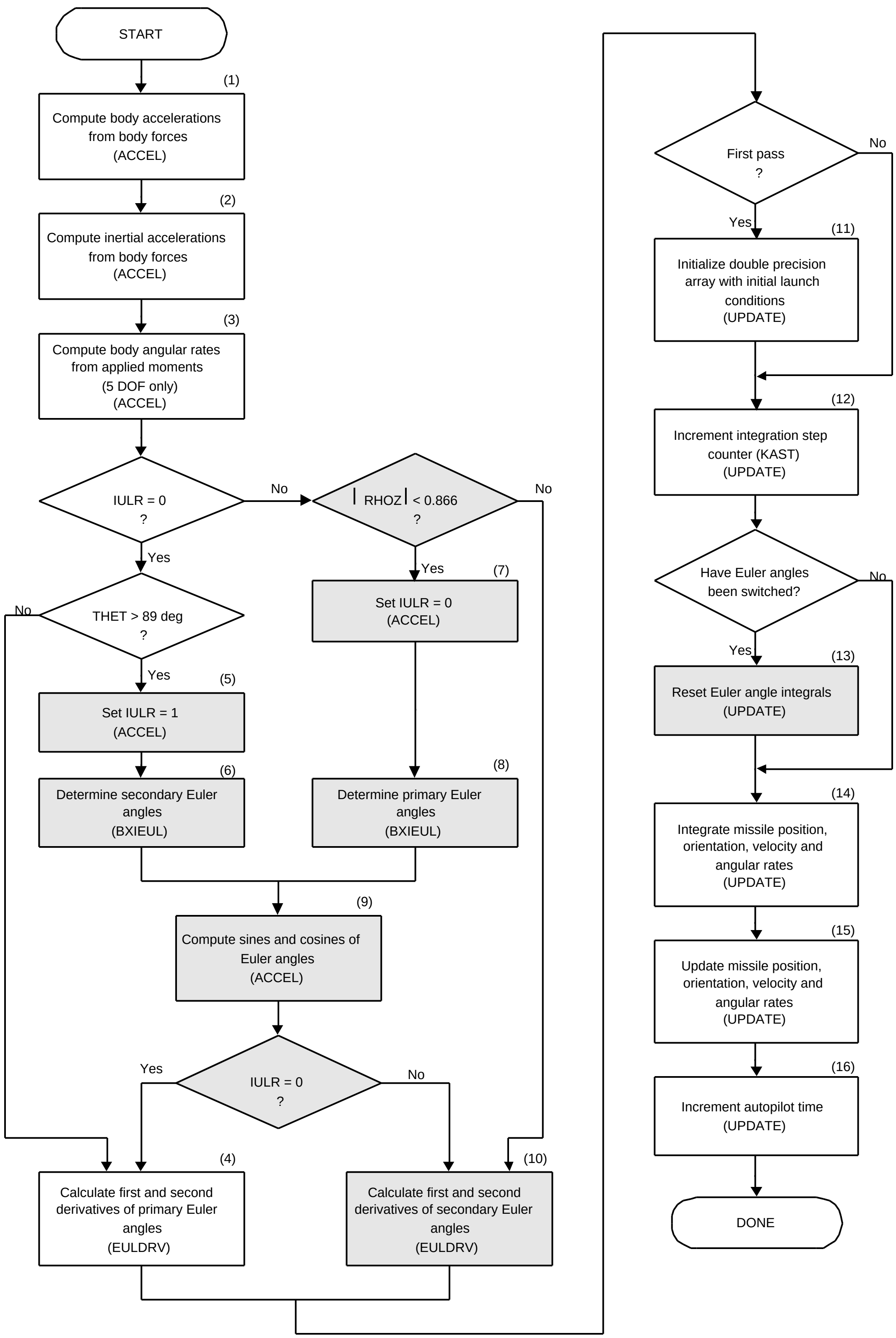

FIGURE 2.41-1. Logical Flow Discrepancies.

2.41.4 Software Test Cases and Results

All software testing was performed by running the entire ESAMS model in debug mode. For these tests, ESAMS was run using a sample input file for the specific missile of interest.

In some cases, a spreadsheet was utilized to assist in confirming complex code calculations involving non-adjacent but related code. Whenever a spreadsheet was used; however, the same values were tested and verified using ESAMS code.

TABLE 2.41-6. Software Test Cases for Missile Movement.

<table>
<tr><th>TEST CASE ID</th><th>TEST CASE DESCRIPTION</th></tr>
<tr><td>41-1</td><td>OBJECTIVE: Check calculation of inertial accelerations (XDDOT, YDDOT and ZDDOT) in ACCEL.

PROCEDURE:

1. Run ESAMS. Set a breakpoint in the ACCEL subroutine at line 116 and deposit the following values for THRUST, FA(1), FA(2), FA(3), FMASS and G.
THRUST = 11000.
FA(1) = 3786.
FA(2) = 2066.
FA(3) = 557.
FMASS = 127.
G = 9.80665
2. Deposit the following values for RHOX, RHOY, RHOZ, PIX, PIY, PIZ, ETAX, ETAY and ETAZ.
<table>
<tr><th></th><th colspan="5">CASES</th></tr>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>RHOX</td><td>1.0</td><td>0.0</td><td>0.0</td><td>-0.633</td><td>0.483</td></tr>
<tr><td>RHOY</td><td>0.0</td><td>1.0</td><td>1.0</td><td>0.1116</td><td>-.8365</td></tr>
<tr><td>RHOZ</td><td>0.0</td><td>0.0</td><td>0.0</td><td>0.766</td><td>-.2588</td></tr>
<tr><td>PIX</td><td>0.0</td><td>-1.0</td><td>-0.7071</td><td>0.4107</td><td>0.7039</td></tr>
<tr><td>PIY</td><td>1.0</td><td>0.0</td><td>0.0</td><td>-0.7904</td><td>0.1951</td></tr>
<tr><td>PIZ</td><td>0.0</td><td>0.0</td><td>0.7071</td><td>0.4545</td><td>0.683</td></tr>
<tr><td>ETAX</td><td>0.0</td><td>0.0</td><td>0.7071</td><td>0.6562</td><td>-0.5209</td></tr>
<tr><td>ETAY</td><td>0.0</td><td>0.0</td><td>0.0</td><td>0.6023</td><td>-0.5121</td></tr>
<tr><td>ETAZ</td><td>1.0</td><td>1.0</td><td>0.7071</td><td>0.4545</td><td>0.683</td></tr>
</table>
3. Use Equations [2.41-4] through [2.41-6] of ASP II to independently calculate the values of XDDOT, YDDOT, and ZDDOT for the values in steps 1 & 2.
4. Note the values of XDDOT at line 116, YDDOT at line 117 and ZDDOT at line 118 the and compare to independently calculated values

VERIFY: ESAMS values match the independently calculated values.

RESULT: OK</td></tr>
</table>

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

<table>
<tr><th>TEST CASE ID</th><th>TEST CASE DESCRIPTION</th></tr>
<tr><td>41-2</td><td>OBJECTIVE: Check calculation of pitch and yaw accelerations (OMEGD(2) and OMEGD(3)) in ACCEL.

PROCEDURE:
1. Run ESAMS. Set a breakpoint in the ACCEL subroutine at line 124 and set IAUTO = 0.
2. Skip to next line of code, verify IF statement operation.
3. Return to line 124 in subroutine ACCEL, set IAUTO = 1 and deposit the following values for TAUA(2), TAUA(3) and AINERT.
<u>TAUA(2)</u> <u>TAUA(3)</u> <u>AINERT</u>
-2.372 1.157 51.328
0.0 2.56 52.701
0.987 0.0 52.886
2.458 -1.033 53.953
4. Use Equations [2.41-13] through [2.41-15] of ASP II to independently calculate the values of OMEGD(2) and OMEGD(3) for the values in step 1.
5. Note the values of OMEGD(2) at line 127 and OMEGD(3) at line 128 and compare to independently calculated values.

VERIFY: ESAMS values match the independently calculated values.

RESULT: OK</td></tr>
<tr><td>41-3</td><td>OBJECTIVE: Check setting of Euler angle switching flag, IULR in ACCEL.

PROCEDURE:
1. Run ESAMS. Set a breakpoint in the ACCEL subroutine at line 134 and deposit the following values for IULR and THET.
IULR = 0
THET = 0.987
2. Step to next line of code and observe.
3. Return to line 134 and deposit the following values for IULR and THET.
IULR = 0
THET = 1.554
4. Step to next line of code and observe.
5. Return to line 134 and deposit the following values for IULR and RHOZ.
IULR = 1
RHOZ = 0.867
6. Step to next line of code and observe.

VERIFY:
1. In step 2, execution proceeds to line 138 with no change in IULR.
2. In step 4, execution proceeds to line 138 with no change in IULR.
3. In step 6, execution proceeds to line 140, the IF statement is satisfied and IULR changes to 0.

RESULT: The comment on line 137 prevents the IF statement on line 136 from ever setting IULR = 1. The result is that IULR will never be set to 1. This is the desired result from the implementation of MDR #1. When IULR is manually set to 1 the IF statement on line 140 operates correctly, setting IULR = 0.</td></tr>
</table>

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

<table>
<tr><th>TEST CASE ID</th><th>TEST CASE DESCRIPTION</th></tr>
<tr><td>41-4</td><td>OBJECTIVE: Check calculation of sines and cosines of the Euler angles in ACCEL.
PROCEDURE:
1. Run ESAMS. Set a breakpoint in the ACCEL subroutine at line 144 and deposit the following values for IULR, IO, PSI, THET and PHI.
IULR = 1
IO = 0
<u>PSI</u> <u>THET</u> <u>PHI</u>
0.0 -1.5708 -1.5708
1.484 -1.0132 -1.0132
2.5428 -0.9736 -0.9736
3.1416 0.0 0.0
4.3633 1.1197 1.1197
5.6723 1.5708 1.5708
2. Independently determine the values of the sines and cosines of the values in step 1.
3. Note the values of SPSI, CPSI, STHT, CTHT, SPHI and CPHI at lines 147-152 and compare to independent values.
VERIFY: ESAMS values match the independently determined values.
RESULT: OK</td></tr>
<tr><td>41-5</td><td>OBJECTIVE: Check calculation of primary Euler angles from secondary matrix terms in BXIEUL.
PROCEDURE:
1. Run ESAMS. Set a breakpoint in the BXIEUL subroutine at line 71 and deposit the following value for IUL. IUL = 0
2. Deposit the following values for RHOZ, RHOY, RHOX, PIZ and ETAZ.
<u>RHOZ</u> <u>RHOY</u> <u>RHOX</u> <u>PIZ</u> <u>ETAZ</u>
-1.0 0.0 -0.573 0.0 -0.573
-0.832 0.0117 0.0 0.0117 0.0
-0.016 -0.627 -0.983 -0.627 -0.983
0.0 -0.52 0.488 -0.52 0.488
0.337 0.377 0.787 0.377 0.787
0.568 0.556 -0.677 0.556 -0.677
1.0 -0.573 0.0 -0.573 0.0
3. Using Equations [2.41-45] , [2.41-47], and [2.41-49] of ASP II, independently calculate the values of THET, PSI and PHI for the values in steps 1-3.
4. Note the values of THET at line 76, PSI at line 77 and PHI at line 78 and compare to independently calculated values
VERIFY: ESAMS values match the independently calculated values.
RESULT: OK</td></tr>
</table>

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-6	OBJECTIVE: Check calculation of secondary Euler angles from primary matrix terms in BXIEUL. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the BXIEUL subroutine at line 71 and deposit the following value for IUL. IUL = 1 2. Deposit the following values for ETAZ, ETAY, ETAX, PIZ and RHOZ. ETAZ / ETAX / ETAY / RHOZ / PIZ -1.0 / 0.0 / -0.573 / 0.0 / -0.573 -0.832 / 0.0117 / 0.0 / 0.0117 / 0.0 -0.016 / -0.627 / -0.983 / -0.627 / -0.983 0.0 / -0.52 / 0.488 / -0.52 / 0.488 0.337 / 0.377 / 0.787 / 0.377 / 0.787 0.568 / 0.556 / -0.677 / 0.556 / -0.677 1.0 / -0.573 / 0.0 / -0.573 / 0.0 3. Using Equations [2.41-39], [2.41-41], and [2.41-43] of ASP II, independently calculate the values of THET, PSI and PHI for the values in step 1-3. 4. Note the values of THET at line 93, PSI at line 94 and PHI at line 95 and compare to independently calculated values VERIFY: ESAMS values match the independently calculated values. RESULT: OK
41-7	OBJECTIVE: Check calculation of the first derivatives of the primary Euler angles in EULDRV. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the EULDRV subroutine at line 80 and deposit the following value for IULR. IULR = 0 2. Step to line 81 and deposit the following values for OMEG(1), OMEG(2), OMEG(3), SPHI, CPHI, CTHT and STHT. OMEG(1) = 0.0 OMEG(2) / OMEG(3) / SPHI / CPHI / CTHT / STHT 2.631 / 0.869 / -0.707 / 0.70720 / 0.321 / 0.9471 -1.075 / -1.776 / -0.321 / 0.9471 / 0.011 / 0.9999 0.0 / -2.113 / 0.011 / 0.9999 / 0.0 / 1.0 0.786 / 0.0 / 0.0 / 1.0 / 0.557 / 0.8305 1.599 / 0.985 / 0.557 / 0.8305 / -0.707 / 0.7072 3. Using Equations [2.41-23] through [2.41-25] of ASP II, independently calculate the values of PSID, THETD and PHID for the values in step 2. 4. Note the values of PSID at line 81, THETD at line 82 and PHID at line 83 and compare to independently calculated values. VERIFY: ESAMS values match the independently calculated values. RESULT: A divide by 0 error occurs when CTHT = 0. There is nothing in the code to prevent this since the switch to alternate Euler angles is inactivated.

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

<table>
<tr><th>TEST CASE ID</th><th>TEST CASE DESCRIPTION</th></tr>
<tr><td>41-8</td><td>OBJECTIVE: Check calculation of the second derivatives of the primary Euler angles in EULDRV.
PROCEDURE:
1. Run ESAMS. Set a breakpoint in the EULDRV subroutine at line 80 and deposit the following value for IULR. IULR = 0
2. Step to line 84 and deposit the following values for OMEGD(1), OMEGD(2), OMEGD(3), SPHI, CPHI, CTHT, STHT, PHID, PSID and THETD.
OMEGD(1) = 0.0
TEST CASES
<table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>OMEGD(2)</td><td>-0.576</td><td>-0.317</td><td>0.0</td><td>0.022</td><td>0.989</td></tr>
<tr><td>OMEGD(3)</td><td>0.144</td><td>-0.111</td><td>-0.988</td><td>0.0</td><td>0.376</td></tr>
<tr><td>SPHI</td><td>0.707</td><td>0.321</td><td>-0.011</td><td>0.0</td><td>0.557</td></tr>
<tr><td>CPHI</td><td>0.7072</td><td>0.9471</td><td>0.9999</td><td>1.0</td><td>0.8305</td></tr>
<tr><td>CTHT</td><td>0.321</td><td>0.011</td><td>0.0</td><td>-0.557</td><td>0.458</td></tr>
<tr><td>STHT</td><td>0.9471</td><td>0.9999</td><td>1.0</td><td>0.8305</td><td>0.8889</td></tr>
<tr><td>PHID</td><td>3.6758</td><td>-1.77607</td><td>-2.113</td><td>0.0</td><td>-0.07262</td></tr>
<tr><td>PSID</td><td>-3.8811</td><td>0.00488</td><td>0.0</td><td>0.0</td><td>0.10269</td></tr>
<tr><td>THETD</td><td>-2.47504</td><td>1.075</td><td>0.0</td><td>0.786</td><td>1.8766</td></tr>
</table>
3. Using Equations [2.41-26] through [2.41-28] of ASP II, independently calculate the values of THETDD, PSIDD and PHIDD for the values in step 2.
4. Note the values of THETDD at line 85, PHIDD at line 86 and PSIDD at line 87 and compare to independently calculated values.
VERIFY: ESAMS values match the independently calculated values.
RESULT: A divide by 0 error occurs when CTHT = 0. There is nothing in the code to prevent this since the switch to alternate Euler angles is inactivated.</td></tr>
</table>

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

<table>
<tr><th>TEST CASE ID</th><th>TEST CASE DESCRIPTION</th></tr>
<tr><td>41-9</td><td>OBJECTIVE: Check for potential overflow in calculation of PSID, PHID, PSIDD and PHIDD in EULDRV.

PROCEDURE:

1. Run ESAMS. Set a breakpoint in the EULDRV subroutine at line 80 and deposit the following value for IULR.
IULR = 0
2. Step to line 81 and deposit the following values for OMEG(1), OMEG(2), OMEG(3), OMEGD(1), OMEGD(2), OMEGD(3), SPHI and CPHI.
OMEG(1) = 0.0
OMEG(2) = 1.536
OMEG(3) = 1.55
OMEGD(1) = 0.0
OMEGD(2) = 0.773
OMEGD(3) = 0.761
SPHI = 0.7071
CPHI = 0.7071
3. Deposit the following values for CTHT and STHT.
<u>CTHT</u> <u>STHT</u> <u>(THETA in degrees)</u>
0.01745 0.99985 89.0
0.00873 0.99996 89.5
0.001745 0.99999 89.9
0.00001745 0.99999 89.999
0.00000174 0.99999 89.9999
4. Use Equations [2.41-23], [2.41-25], [2.41-26], and [2.41-28] of ASP II to independently calculate the values of PSID, PHID, PSIDD and PHIDD for the values in steps 2 and 3.
5. Note the values of PSID at line 81, PHID at line 83, PHIDD at line 86 and PSIDD at lin 87 and compare to independently calculated values.

VERIFY:

1. Potential overflow.
2. ESAMS values match the independently calculated values.

RESULT: When CTHT equals a very small number, the values for PSID, PSIDD, PHID and PHIDD become unreasonably large, causing extremely unrealistic missile roll. This is a result of the “short-term solution” that eliminates the switch to alternate Euler angles.</td></tr>
</table>

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-10	OBJECTIVE: Check calculation of PSID when PSI is outside of 0 to 2π range in EULDRV. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the EULDRV subroutine at line 80 and deposit the following value for IULR. IULR= 0 2. Step to line 88 and deposit the following values for PSI and PSID(on the right of the equal sign). PSI / PSID -0.21 / 5.67 0.0 / 5.67 4.376 / 5.67 6.28318 / 5.67 6.3 / 5.67 3. Use Equation [2.41-23] of ASP II to independently calculate the value of PSID for the values in step 2. 4. Note the value of PSID at line 89 or 92 and compare to independently calculated values. VERIFY: ESAMS values match independently calculated values. RESULT: OK
41-11	OBJECTIVE: Check calculation of the first derivatives of the secondary Euler angles in EULDRV. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the EULDRV subroutine at line 80 and deposit the following value for IULR. IULR = 1 2. Step to line 95 and deposit the following values for OMEG(1), OMEG(2), OMEG(3), SPHI, CPHI, CTHT and STHT. OMEG(1) = 0.0 OMEG(2) / OMEG(3) / SPHI / CPHI / CTHT / STHT 2.631 / 0.869 / -0.707 / 0.7072 / 0.321 / 0.9471 -1.075 / -1.776 / -0.321 / 0.9471 / 0.011 / 0.9999 0.0 / -2.113 / 0.011 / 0.9999 / 0.0 / 1.0 0.786 / 0.0 / 0.0 / 1.0 / 0.557 / 0.8305 1.599 / 0.985 / 0.557 / 0.8305 / 0.707 / -0.7072 3. Use Equations [2.41-32] through [2.41-34] to independently calculate the values of PSID, THETD and PHID for the values in step 2. 4. Note the values of PSID at line 95, THETD at line 96 and PHID at line 97 and compare to independently calculated values. VERIFY: ESAMS values match the independently calculated values. RESULT: OK

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-12	OBJECTIVE: Check calculation of the second derivatives of the secondary Euler angles in EULDRV. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the EULDRV subroutine at line 80 and deposit the following value for IULR. IULR = 1 2. Step to line 98 and deposit the following values for OMEGD(1), OMEGD(2), OMEGD(3), SPHI, CPHI, CTHT, STHT, PHID, PSID and THETD. OMEGD(1) = 0.0 TEST CASES <table><tr><th></th><th><u>1</u></th><th><u>2</u></th><th><u>3</u></th><th><u>4</u></th><th><u>5</u></th></tr><tr><td>OMEGD(2)</td><td>-0.576</td><td>-0.317</td><td>0.0</td><td>0.022</td><td>0.989</td></tr><tr><td>OMEGD(3)</td><td>0.144</td><td>-0.111</td><td>-0.988</td><td>0.0</td><td>0.376</td></tr><tr><td>SPHI</td><td>0.707</td><td>0.321</td><td>-0.011</td><td>0.0</td><td>0.557</td></tr><tr><td>CPHI</td><td>0.7072</td><td>0.9471</td><td>0.9999</td><td>1.0</td><td>0.8305</td></tr><tr><td>CTHT</td><td>0.321</td><td>0.011</td><td>0.0</td><td>-0.557</td><td>0.458</td></tr><tr><td>STHT</td><td>0.9471</td><td>0.9999</td><td>1.0</td><td>0.8305</td><td>0.8889</td></tr><tr><td>PHID</td><td>3.6758</td><td>-1.77607</td><td>-2.113</td><td>0.0</td><td>-0.07262</td></tr><tr><td>PSID</td><td>-3.8811</td><td>0.00488</td><td>0.0</td><td>0.0</td><td>0.10269</td></tr><tr><td>THETD</td><td>-2.47504</td><td>1.075</td><td>0.0</td><td>0.786</td><td>1.8766</td></tr></table>3. Use Equations [2.41-35] through [2.41-37] of ASP II to independently calculate the values of THETDD, PSIDD and PHIDD for the values in step 2. 4. Note the values of THETDD at line 99, PHIDD at line 100 and PSIDD at line 101 and compare to independently calculated values. VERIFY: ESAMS values match the independently calculated values. RESULT: OK

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-13	OBJECTIVE: Check initialization of double precision array and local Euler angle flag, IULRO, with initial values in UPDATE PROCEDURE: 1. Run ESAMS. Set a breakpoint in the UPDATE subroutine at line 99 and verify this IF statement is true on the first pass. 2. Step to line 102 and verify value of DXI(1) is set equal to initial value of X and that DXI(1) is of double precision accuracy. 3. Step to line 103 and verify value of DXI(2) is set equal to initial value of Y and that DXI(2) is of double precision accuracy. 4. Step to line 104 and verify value of DXI(3) is set equal to initial value of Z and that DXI(3) is of double precision accuracy. 5. Step to line 107 and verify value of DXI(4) is set equal to initial value of XDOT and that DXI(4) is of double precision accuracy. 6. Step to line 108 and verify value of DXI(5) is set equal to initial value of YDOT and that DXI(5) is of double precision accuracy. 7. Step to line 109 and verify value of DXI(6) is set equal to initial value of ZDOT and that DXI(6) is of double precision accuracy. 8. Step to line 112 and verify value of DXI(7) is set equal to initial value of PSI and that DXI(7) is of double precision accuracy. 9. Step to line 113 and verify value of DXI(8) is set equal to initial value of THET and that DXI(8) is of double precision accuracy. 10. Step to line 114 and verify value of DXI(9) is set equal to initial value of PHI and that DXI(9) is of double precision accuracy. 11. Step to line 117 and verify value of DXI(10) is set equal to 0.0 and that DXI(10) is of double precision accuracy. 12. Step to line 118 and verify value of DXI(11) is set equal to 0.0 and that DXI(11) is of double precision accuracy. 13. Step to line 119 and verify value of DXI(12) is set equal to 0.0 and that DXI(12) is of double precision accuracy. 14. Step to line 122 and verify value of IULRO is set equal to IULR. VERIFY: ESAMS initializes double precision array and local Euler angle flag, IULRO, with correct initial values. RESULT: OK

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-14	OBJECTIVE: Check resetting of Euler angles after a switch in UPDATE. PROCEDURE: 1. Run ESAMS. Allow ESAMS to cycle through several times then set a breakpoint in the UPDATE subroutine at line 129 and deposit the following values for IULRO and IULR. IULRO = 0 IULR = 1 2. Step to line 132 and verify value of DXI(7) is set equal to current value of PSI and that DXI(7) is of double precision accuracy. 3. Step to line 133 and verify value of DXI(8) is set equal to current value of THET and that DXI(8) is of double precision accuracy. 4. Step to line 134 and verify value of DXI(9) is set equal to current value of PHI and that DXI(9) is of double precision accuracy. 5. Step to line 136 and verify that IULRO is set equal to 1. VERIFY: ESAMS resets Euler angles with current values. RESULT: OK
41-15	OBJECTIVE: Check calculation of intermediate and final position coordinates (DXI(1), DXI(2) and DXI(3) and of X, Y and Z) in UPDATE. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the UPDATE subroutine at line 140 and deposit the following values for DTA, DXI(1), DXI(2), DXI(3), XDOT, YDOT, ZDOT, XDDOT, YDDOT and ZDDOT. DTA 0.02 DXI(1) 5437.229 DXI(2) -845.13 DXI(3) 2227.88 XDOT -455.21 YDOT -3.118 ZDOT 32.866 XDDOT -5.779 YDDOT -0.022 ZDDOT 1.038 2. Use Equations [2.41-51] through [2.41-53] of ASP II to independently calculate the new values of DXI(1), DXI(2) and DXI(3) for the values in step 1. 3. Note the values of DXI(1) at line 140, DXI(2) at line 141 and DXI(3) at line 142 and compare to independently calculated values. 4. Step to line 156 and verify value of X is set equal to current value of DXI(1). 5. Step to line 157 and verify value of Y is set equal to current value of DXI(2). 6 Step to line 158 and verify value of Z is set equal to current value of DXI(3). VERIFY: 1. ESAMS values match the independently calculated values. 2. Position variables X, Y and Z are updated correctly. RESULT: OK

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-16	OBJECTIVE: Check calculation of intermediate and final velocity components (DXI(4), DXI(5) and DXI(6) and XDOT, YDOT and ZDOT) in UPDATE. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the UPDATE subroutine at line 144 and deposit the following values for DTA, DXI(4), DXI(5), DXI(6), XDDOT, YDDOT and ZDDOT. DTA 0.02 DXI(4) -347.881 DXI(5) 58.002 DXI(6) -10.023 XDDOT -3.772 YDDOT 2.228 ZDDOT -0.9979 2. Use Equations [2.41-58] through [2.41-60] of ASP II to independently calculate the values of DXI(4), DXI(5) and DXI(6) for the values in step 1. 3. Note the values of DXI(4) at line 144, DXI(5) at line 145 and DXI(6) at line 146 and compare to independently calculated values 4. Step to line 160 and verify value of XDOT is set equal to current value of DXI(4). 5. Step to line 161 and verify value of YDOT is set equal to current value of DXI(5). 6. Step to line 162 and verify value of ZDOT is set equal to current value of DXI(6). VERIFY: 1. ESAMS values match the independently calculated values. 2. Inertial velocity variables XDOT, YDOT and ZDOT are updated correctly. RESULT: OK

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-17	OBJECTIVE: Check calculation of intermediate and final Euler angles (DXI(7), DXI(8) and DXI(9) and PSI, THET and PHI) in UPDATE. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the UPDATE subroutine at line 148 and deposit the following values for DTA, DXI(7), DXI(8), DXI(9), PSID, THETD, PHID, PSIDD, THETDD and PHIDD. DTA 0.02 DXI(7) 3.7298 DXI(8) 0.8766 DXI(9) 0.6677 PSID -1.0153 THETD 0.9029 PHID -0.2216 PSIDD -0.5527 THETDD 0.4531 PHIDD -0.1118 2. Use Equations [2.41-54] through [2.41-56] of ASP II to independently calculate the values of DXI(7), DXI(8) and DXI(9) for the values in step 1. 3. Note the values of DXI(7) at line 148, DXI(8) at line 149 and DXI(9) at line 150 and compare to independently calculated values. 4. Step to line 164 and verify value of PSI is set equal to current value of DXI(7). 5. Step to line 165 and verify value of THET is set equal to current value of DXI(8). 6. Step to line 166 and verify value of PHI is set equal to current value of DXI(9). VERIFY: 1. ESAMS values match the independently calculated values. 2. Euler angles PSI, THET and PHI are updated correctly. RESULT: OK

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-18	OBJECTIVE: Check calculation of intermediate and final body angular rates (DXI(10), DXI(11) and DXI(12) and OMEG(1), OMEG(2) and OMEG(3)) in UPDATE. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the UPDATE subroutine at line 152 and deposit the following values for DTA, DXI(10), DXI(11), DXI(12), OMEGD(1), OMEGD(2) and OMEGD(3). DTA 0.02 DXI(10) 0.9983 DXI(11) 0.4474 DXI(12) -2.7381 OMEGD(1) 0.0 OMEGD(2) 0.3338 OMEGD(3) -0.9078 2. Use Equations [2.41-61] through [2.41-63] of ASP II to independently calculate the values of DXI(10), DXI(11) and DXI(12) for the values in step 1. 3. Note the values of DXI(10) at line 152, DXI(11) at line 153 and DXI(12) at line 154 and compare to independently calculated values 4. Step to line 168 and verify value of OMEG(1) is set equal to current value of DXI(10). 5. Step to line 169 and verify value of OMEG(2) is set equal to current value of DXI(11). 6. Step to line 170 and verify value of OMEG(3) is set equal to current value of DXI(12). VERIFY: 1. ESAMS values match the independently calculated values. 2. Body rotation rates OMEG(1), OMEG(2) and OMEG(3) are updated correctly. RESULT: OK
41-19	OBJECTIVE: Check incrementing of time step counters for missile integration and autopilot (KAST and ATIME) in UPDATE. PROCEDURE: 1. Run ESAMS. Set a breakpoint in the UPDATE subroutine at line 126 and observe value of KAST on first pass through. On second pass through note value of KAST again. 2. Set a breakpoint in the UPDATE subroutine at line 172 and observe value of ATIME and DTA. On next pass through, note value of ATIME again. VERIFY: ESAMS correctly increments KAST and ATIME by 1. RESULT: OK

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-20	OBJECTIVE: Check value of missile heading (PSI) when it crosses 0° from both directions. PROCEDURE: 1. Run ESAMS with missile system of interest using initial target coordinates, velocity and heading listed below to force PSI to exceed 2π radians. X position = 2000.0 Y position = -2000.0 Z position = 2500.0 Velocity = 250.0 Heading = 90.0 2. Set a breakpoint in the UPDATE subroutine to examine PSI at line 164 when PSI is transitioning from 2π to 0 radians. 3. Repeat step 1 using initial target coordinates, velocity and heading listed below to force PSI to drop below 0 radians. X position = 2000.0 Y position = 2000.0 Z position = 2500.0 Velocity = 250.0 Heading = 270.0 4. Set a breakpoint in the UPDATE subroutine to examine PSI at line 164 when PSI is transitioning from 0 to 2π radians. VERIFY: ESAMS confines PSI to within 0 to 2π radians. RESULT: In the first scenario, the target is flying north on a line 2000 meters from the missile site. The missile is launched at approximately 330° heading and begins curving to the north to track the target, passing through 360° heading. The second scenario is the opposite of the first with the target flying south. During these runs, the value of PSI was found to exceed 2π radians in the first scenario and drop below zero radians in the second scenario for one iteration before being corrected.

TABLE 2.41-6. Software Test Cases for Missile Movement. (Contd.)

TEST CASE ID	TEST CASE DESCRIPTION
41-21	OBJECTIVE: Test overall effect of “short-term fix” from MDR by checking missile trajectory information following missile pitch exceeding 90°. PROCEDURE: 1. Run ESAMS with missile system of interest using initial target coordinates, velocity and heading listed below. X position = 4800.0 Y position = 0.0 Z position = 2500.0 Velocity = 295.0 Heading = 180.0 2. Examine trajectory information from ESAMS output files. VERIFY: ESAMS maintains trajectory continuity after pitch exceeds 90°. RESULT: In this scenario, the missile first approaches the target head-on flying at 360° heading. Due to the target’s speed the missile must pitch over and fly in the opposite direction (i.e., 180° heading) to continue to close on the target from behind. During this pitch over maneuver the following trajectory information was observed: 1. Missile pitch had increased to 120° when run was completed, indicating the missile is flying “on its back.” Pitch should probably be 60° after the missile rolls to correct itself (or simply maintain its initial roll artificially) 2. The value for roll after pitching over was -4275°, which is meaningless. (This large value is a result of overflow caused by θ’s near 90°) 3. Before the missile pitched over, its heading was 360° (due east). After pitching over, heading remained at 360° and did not correct to 180°.

2.41.5 Conclusions and Recommendations

2.41.5.1 Code Discrepancies

As discussed previously, a recent Model Deficiency Report (ESAMS 2.6.2 Street MDR #1) directed that a critical line of code be commented out. The result of this action was as expected; the use of alternate Euler angles is excluded. Thus, all the code which deals with switching to and using alternate Euler angles is inaccessible. The purpose of having a secondary set of Euler angles is to alleviate a potential for a singularity condition and possible overflow problems. Because secondary Euler angles are not accessible, this potential problem is now probable. Even if overflow or division by zero does not occur, tests indicate that the current coding can produce incorrect or nonsensical missile trajectory and orientation information when the missile has transitioned a near-vertical orientation. The effect of this questionable trajectory data on target closure is not apparent at this time but is of concern. The long-term solution recommended by the MDR is to modify the appropriate sections of the model to correctly implement alternate Euler angles. Whatever the solution, the integrity of missile trajectory data after the missile has transitioned a near-vertical orientation must be of prime concern.

One other minor discrepancy was uncovered. The code allows PSI, the heading, to be outside the specified limits of $[0, 2\pi)$ for one iteration. The recommended solution is to check PSI as soon as it is set and, if necessary, convert it to the equivalent value within the limits.

2.41.5.2 Code Quality and Internal Documentation

Except for the discrepancies listed above, code quality is generally good. The design elements were logically presented with algorithm sequence and code separation enhancing ease of comprehension. Some of the internal documentation was incomplete or incorrect. Additional comments and corrections are recommended.

2.41.5.3 External Documentation

There is no external documentation for ESAMS 2.6.2. Therefore, the external documentation for ESAMS 2.5 was used. This documentation was adequate in the development of determining accelerations and integrating to find the position and velocity parameters over time. The development and presentation of alternate Euler angles and transitioning between Euler angles was insufficient and inadequately referenced.